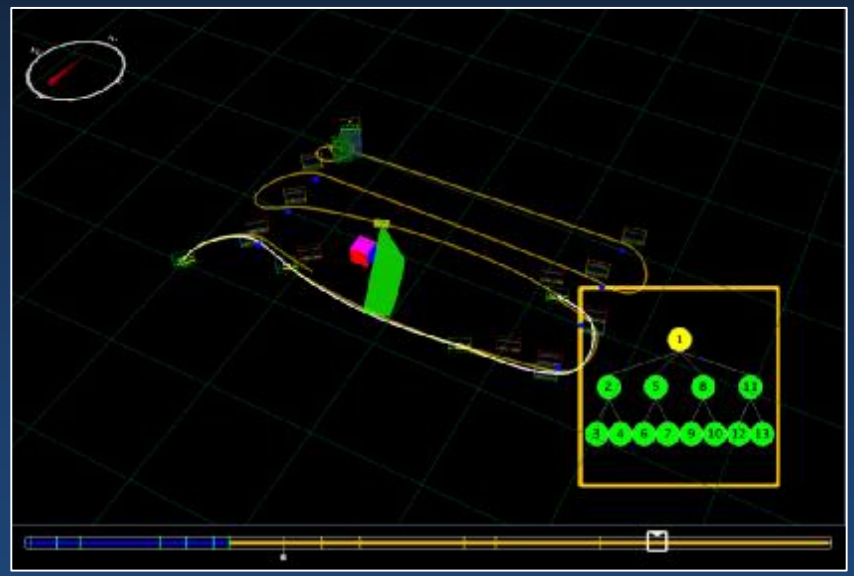
Triton: A System for Enhanced Operator awareness during Autonomous Underwater Missions with Minimal Communication



Christopher C. Sotzing
Nick Johnson
David M. Lane

Ocean Systems Laboratory

School of Engineering & Physical Sciences
Heriot-Watt University, Edinburgh

ccs1@hw.ac.uk

Contents

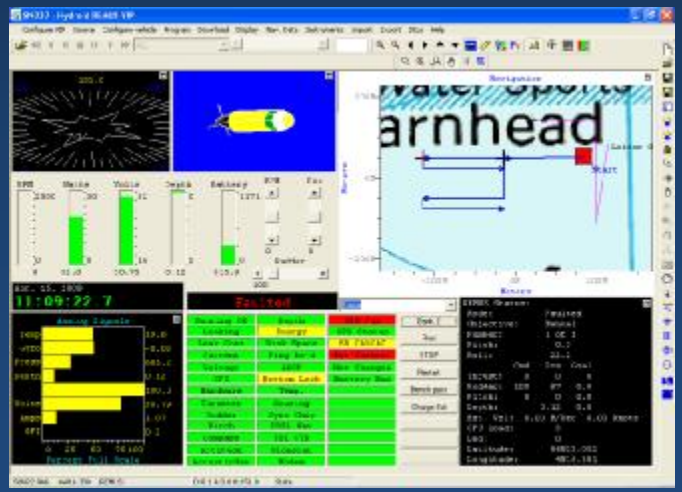
- Motivation
- System Design
- Functionality
- Results
- Conclusions

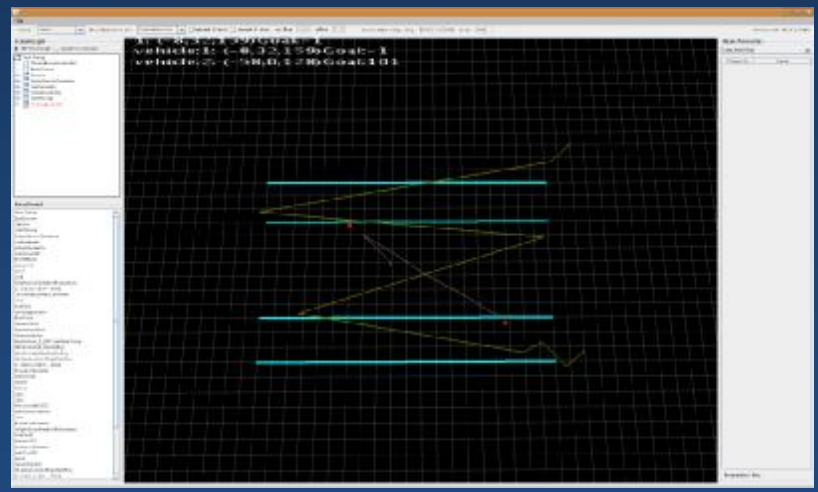
Motivation: Problem Statement

- Operator trust a major challenge in autonomous robotics
- Communication deficient environments (underwater) offer sparse, infrequent updates
  - Information sent by vehicles is simple and often out of date upon reception
  - Users can be “in the dark” for long periods of time
- This research
  - Use prediction, knowledge of vehicle systems to “fill in the blanks” between vehicle updates
  - Keep the operator in the loop and maximise situational awareness

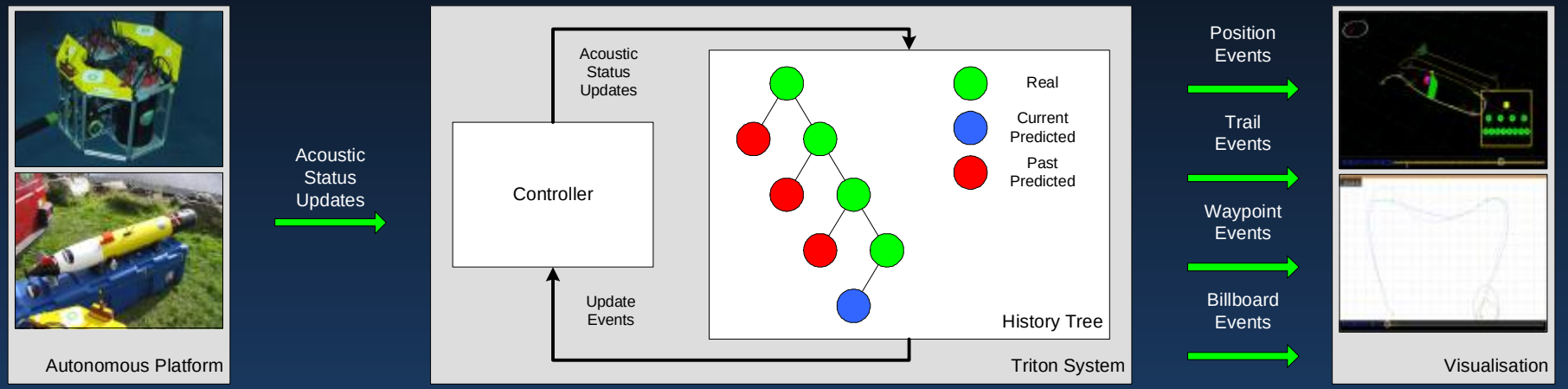
Motivation: State of the Art

- Current topside systems only plot vehicle updates
  - Underlay mission plan
  - Simple, no on-board intelligence
  - Low situational awareness



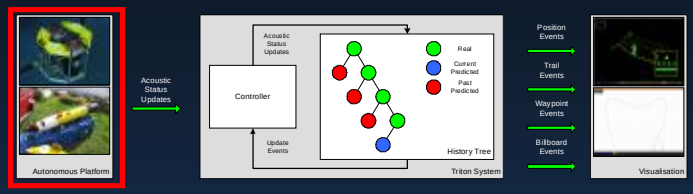


System Design



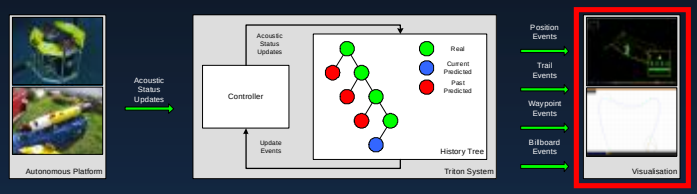
- System made up of three main components:
  - Real platform
  - Triton system
  - Visualisation system
- Supports various vehicle comms (acoustic, wifi, etc.)
- Communication with visualisation via events

System Design: Real Platform

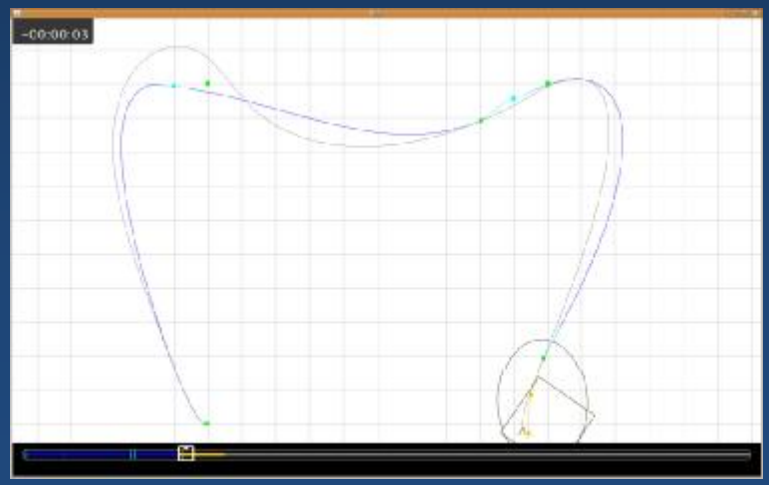


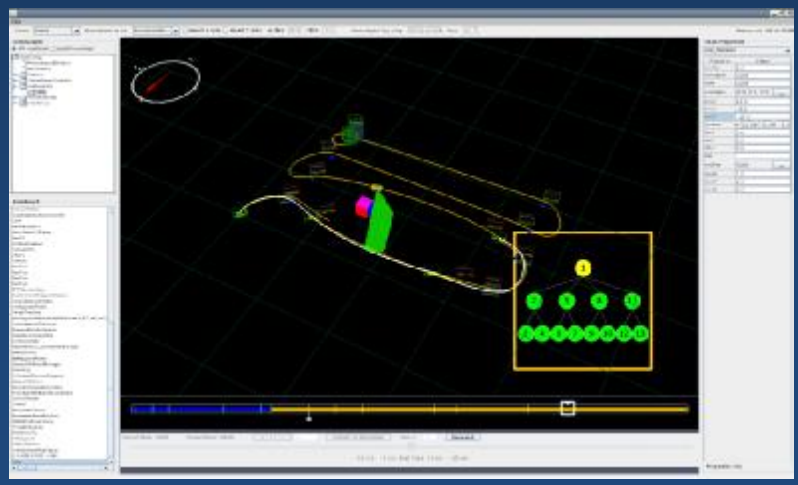
- System can monitor any autonomous robot
  - Mission plan & executive must implement necessary interfaces
- Study utilised DELPHÍS execution system (DTC AA012)
  - Demonstrated on REMUS, Nessie III, RAUVER AUVs
  - Distributed multi-agent based multi-vehicle coordination
  - Blackboard-based hierarchical mission model (BIIMAPS)

System Design: Visualisation

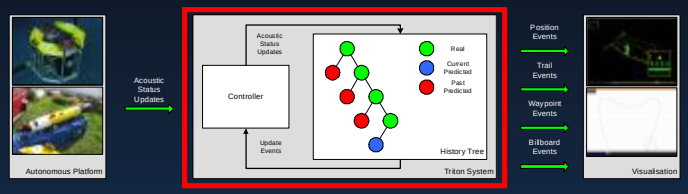


- Any visualisation can be used
  - Must implement the correct set of events (position, trail, waypoint billboard)
- Two visualisation systems demonstrated in this study





System Design: Triton



- Tree based simulation system
  - Real, past predicted, current predicted
  - Real nodes created when updates are received
- Receives updates from vehicle, outputs events to visualisation system
- Able to roll missions forward and backward
  - Always calculating 10 minutes into the future from the current point

Functionality

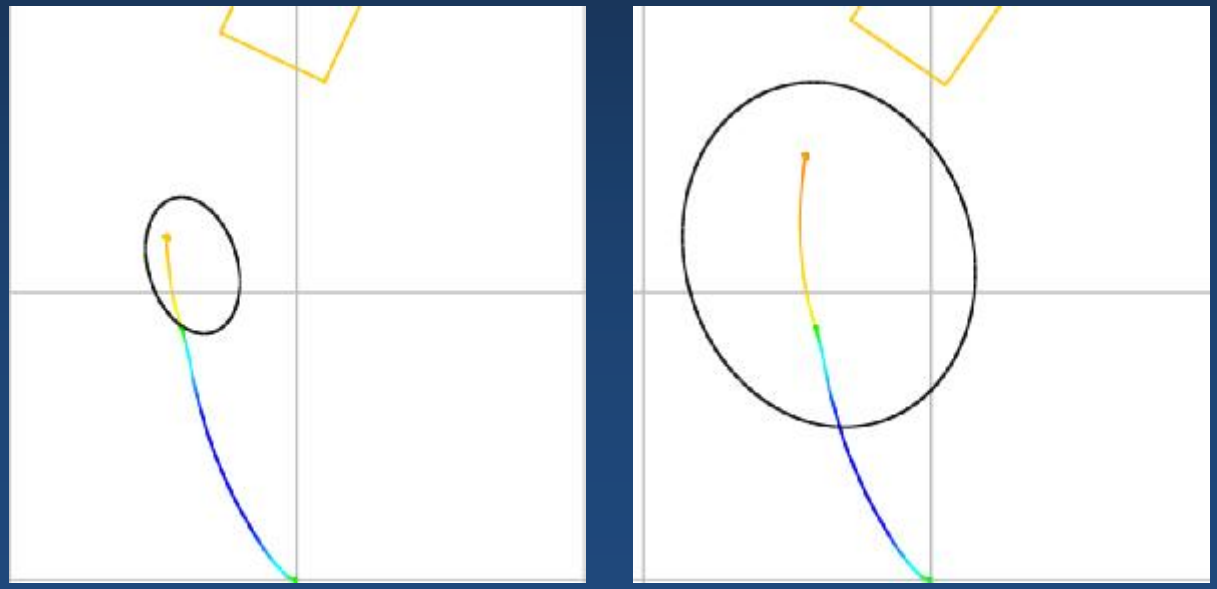
- Vehicle prediction
- Simulated/real vehicle synchronisation
- Estimation of vehicle position uncertainty
- Fault simulation
- Path extrapolation
- “Timeline” functionality
- Graphical User Interface (GUI)

Functionality: Prediction/Synchronisation

- System can predict vehicle position and state using a simulated instance of the vehicle
- Information from vehicle is sent acoustically and simulation is updated/rolled back
- Acoustic broadcast message from DELPHÍS was utilised
  - Vehicle position
  - Vehicle attitude/speed (if bandwidth allows)
  - Ordered list of completed goals
  - Current goal
  - List of potential targets

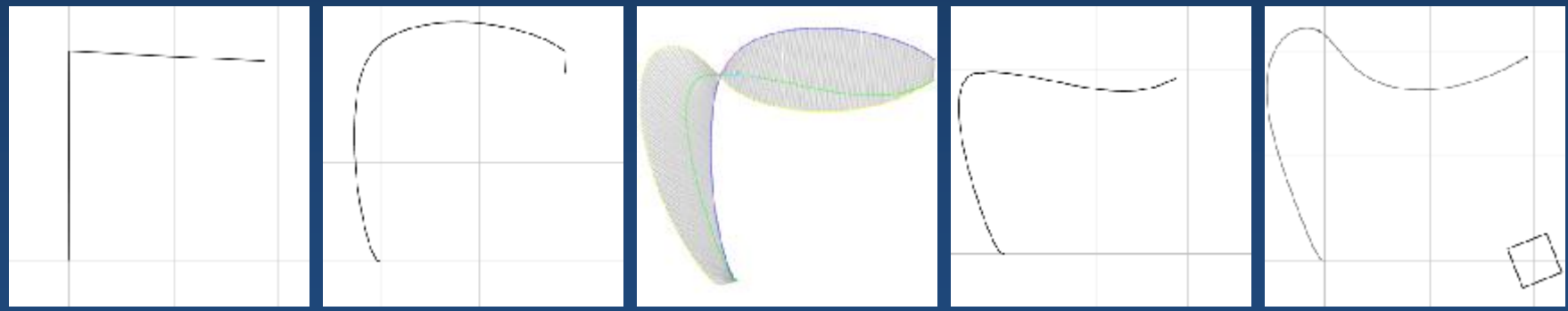
Functionality: Uncertainty/Faults

- Uncertainty represented by a growing spheroid detailing all the possible locations of the vehicle
- Simulated faults can be added throughout the mission to see their possible affect



Functionality: Path Extrapolation

- Most likely vehicle path can be extrapolated from vehicle update to last known update
- 4 step process
- Simple solution: good enough estimation, low overhead

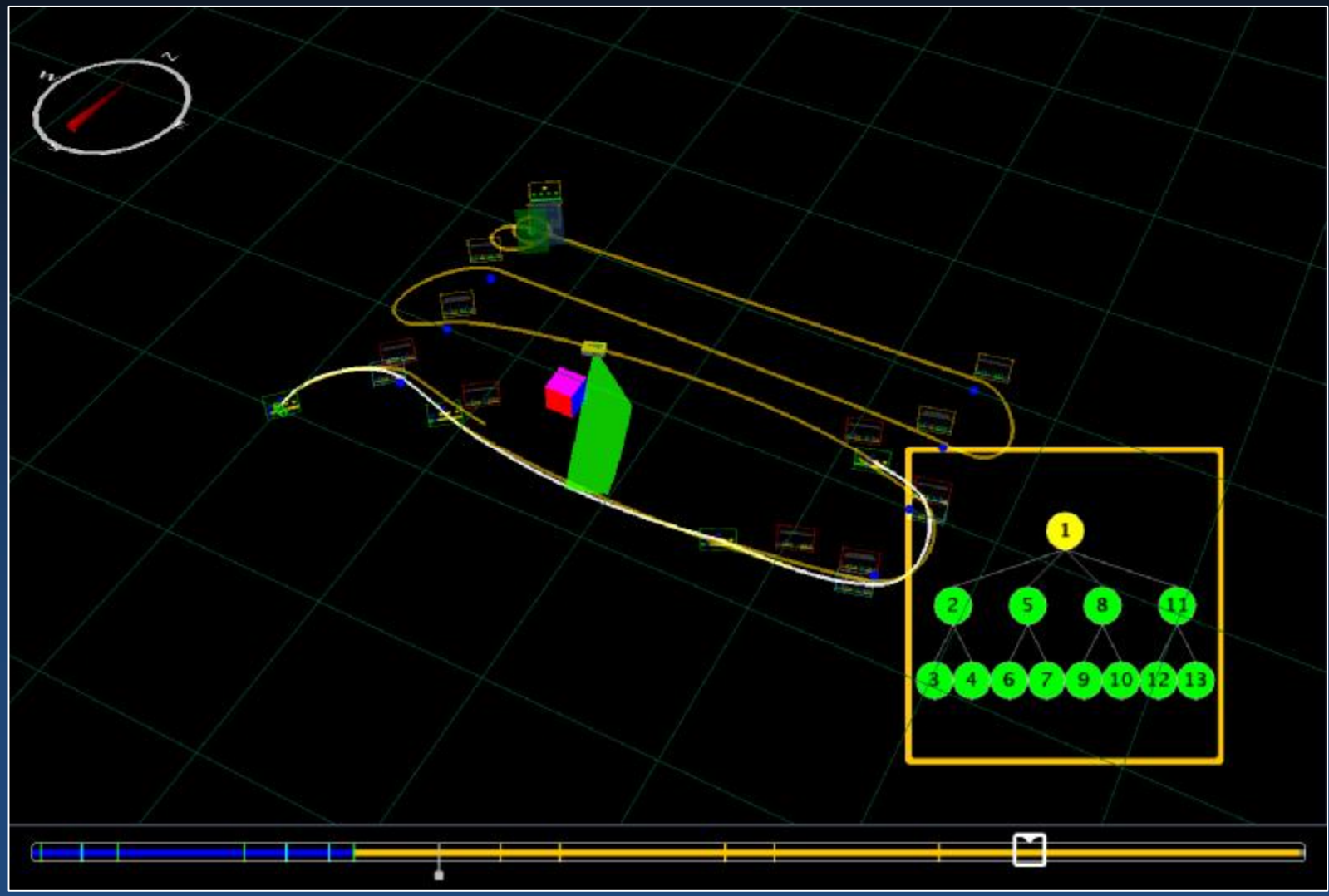


Granular Forward Backward Hybrid Real

Functionality: Graphical User Interface (GUI)

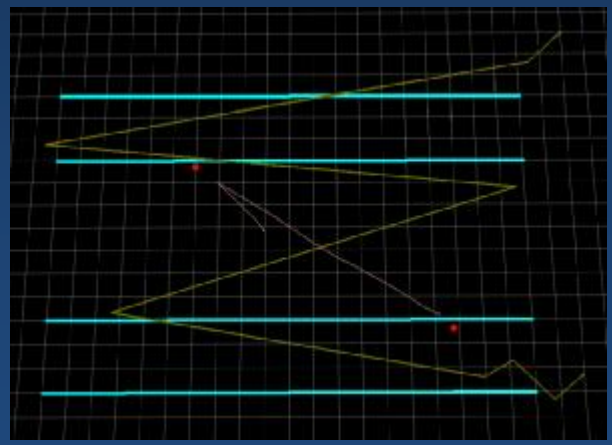
- Timeline
  - Draggable into mission future, past
- Vehicle path
  - Coloured to denote real/predicted path
- Billboards
  - Display mission events
- Heads Up Display (HUD)
  - Compass, time remaining, current mission state
- Waypoints
  - Coloured to denote status
- Uncertainty spheroid

Functionality: Graphical User Interface (GUI)

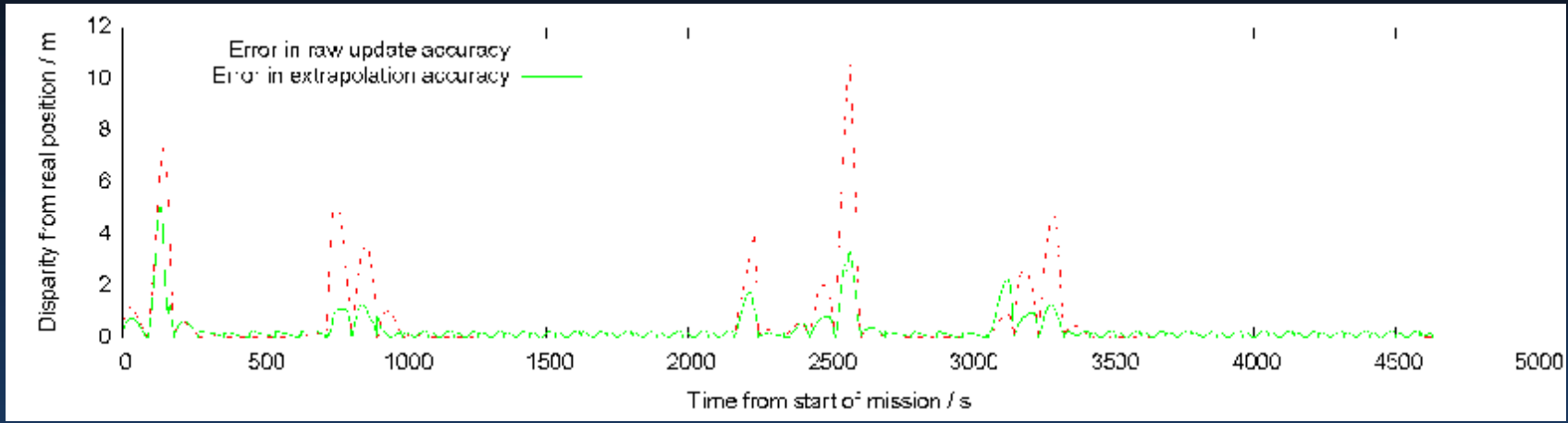


Results

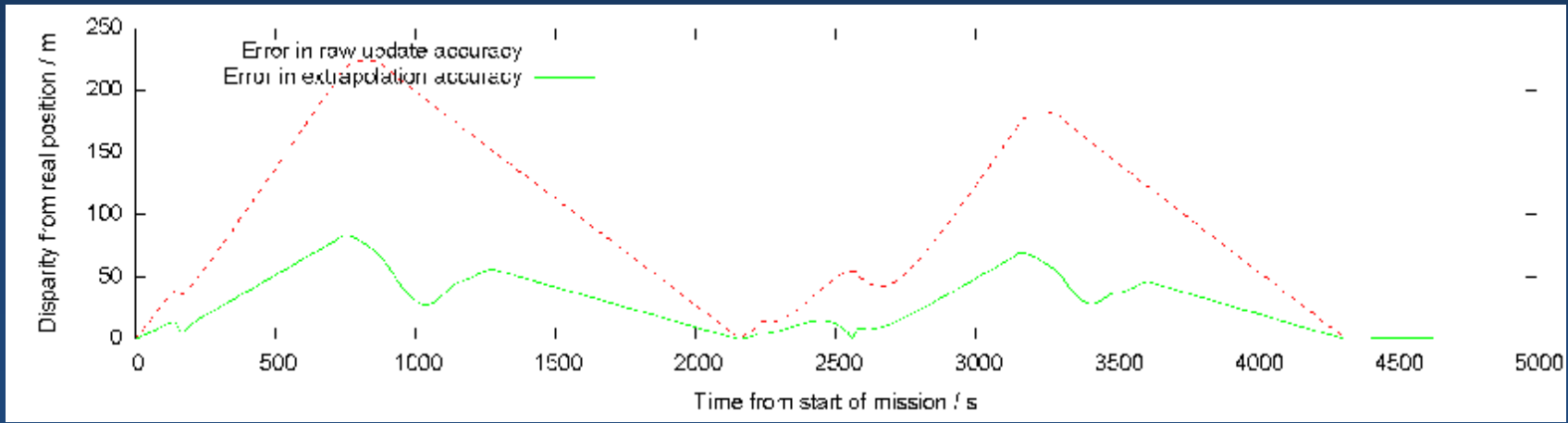
- Survey mission (4 leg lawnmower)
- Varying time between vehicle updates
  - 1.5 - 48 minute intervals
  - Simulated water current
- Missions compared to state of the art (“jagged line”)
  - *Accuracy*: how well the vehicle history matches that of the real vehicle
  - *Honesty*: how well the system state matches that of the real vehicle



Results: Accuracy

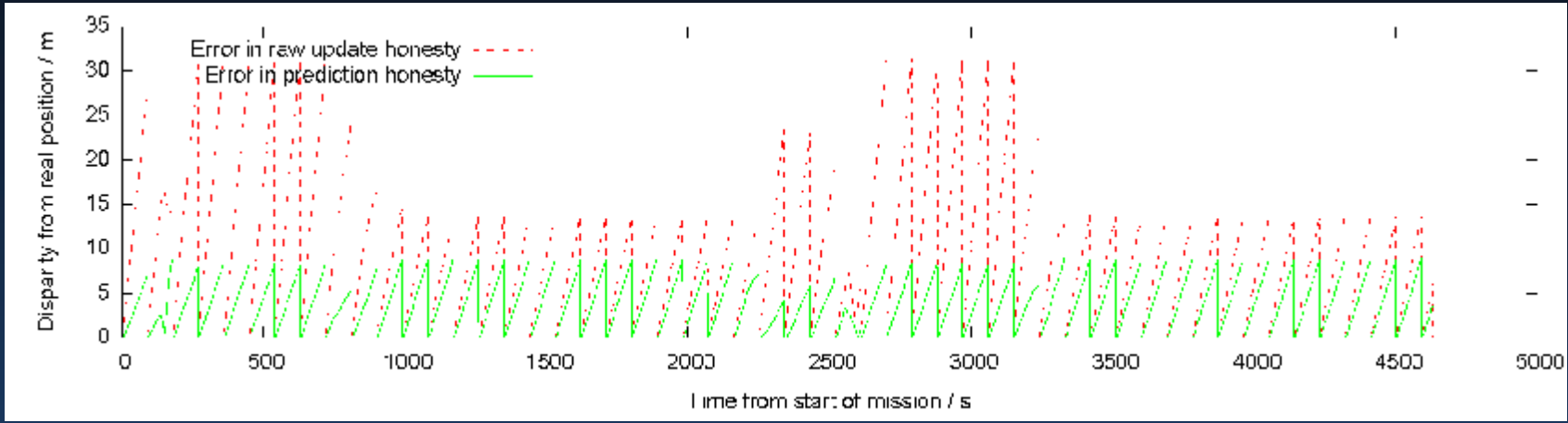


1.5 minute interval

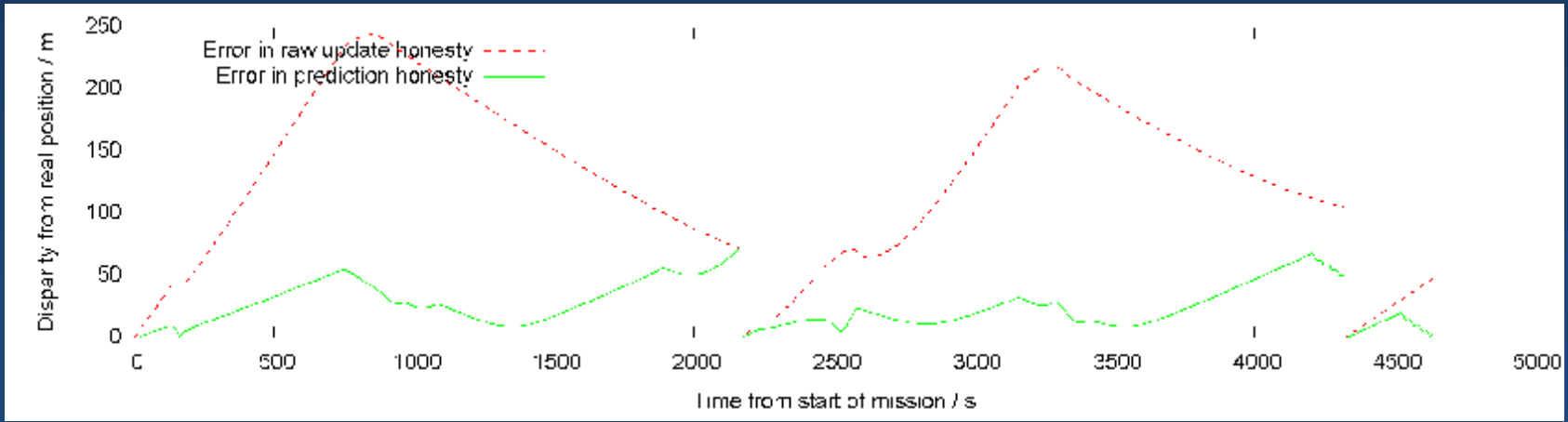


36 minute interval

Results: Honesty

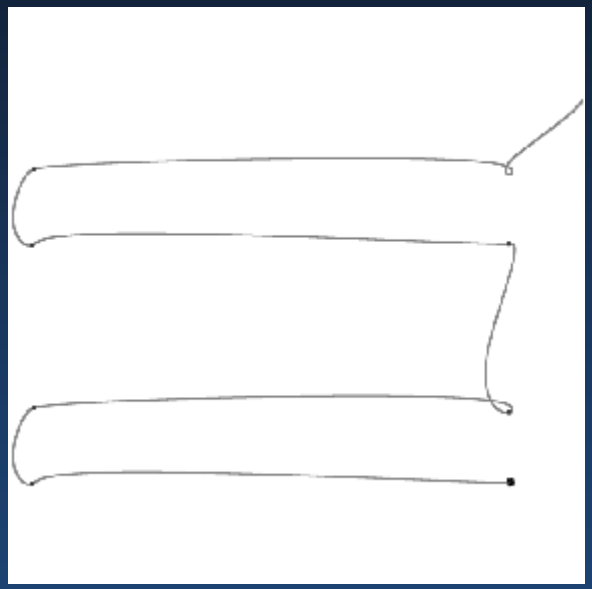


1.5 minute interval

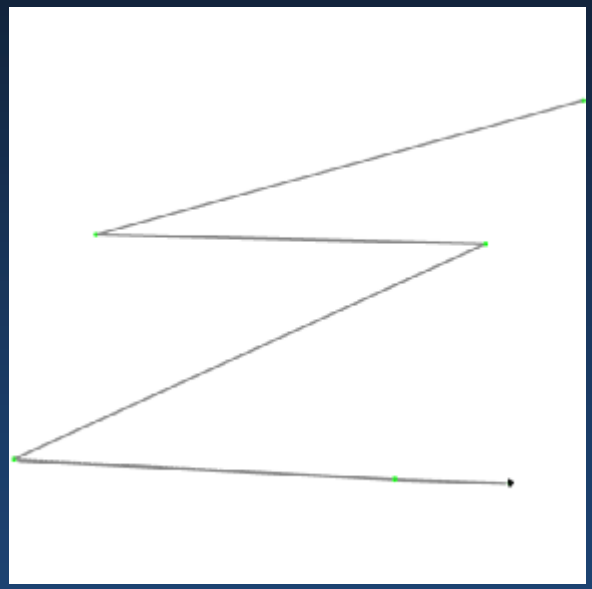


36 minute interval

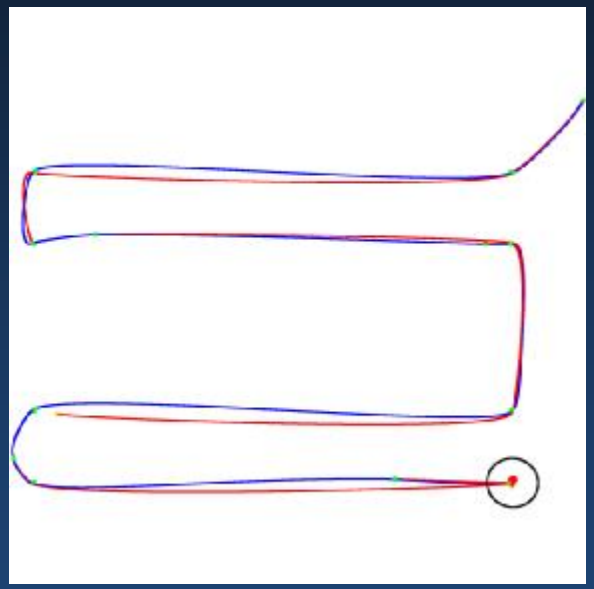
Results: Paths



Real vehicle path



“Zig Zag” updates



Synthetic path

Conclusion

- Current mission monitoring software suites for autonomous underwater vehicles lack sufficient detail to keep operators in the loop
- This work shown to “fill in the blanks” between sparse updates
  - Maximise situational awareness
  - Allow for more accurate mission decisions to be made
- Demonstration to follow in [SEAS DTC Conference venue and time]

Thank You